Before The Board of Patent Appeals & Interferences

For: FAILURE-RESPONSE SIMULATOR FOR COMPUTER CLUSTERS			
Applicant: Jonathan Paul PATRIZIO et al.		Attorney Docket No.: 200314241-1	
Serial No.: 10/767,524 (5514)	Filed: January 29, 2004	Art Unit: 2123	Examiner: Eun Hee CHUNG

Director For Patents
PO Box 1450
Alexandria, VA 22313-1450

Appeal Brief
(Identification Page)

This is an appeal to the Board of Patent Appeals and Interferences from the Final Office Action mailed 2008-May-02 in the above-identified patent application. A Notice of Appeal was filed 2008-Jun-11 (earlier today).

Respectfully submitted
Jonathan Paul PATRIZIO et al.
by

Clifton L. Anderson
Reg. No. 30,989
(408) 257-6070

TABLE OF CONTENTS

REAL PARTY IN INTEREST

The real parties in interest are

Hewlett-Packard Company, a Delaware corporation; and

Hewlett-Packard Development Company, L.P., a Texas limited partnership and wholly owned affiliate of Hewlett-Packard Company, and assignee of record of the Appellants' rights.

RELATED APPEALS AND INTERFERENCES

None.

There are no related appeals or interferences.

STATUS OF CLAIMS

Claims 1-16 are pending in the application.

Claims 1-16 are rejected.

The rejections of Claims 1-16 are being appealed.

STATUS OF AMENDMENTS

All amendments have been entered. There are no unentered amendments.

SUMMARY OF CLAIMED SUBJECT MATTER

Summary Overview

Some computer clusters can reconfigure themselves to continue a mission despite a failure that would halt the mission in the absence of such reconfiguration. The present invention provides a failure-response simulator for evaluating the failure-response of a cluster configuration. The failure-response simulator accepts pre-failure configurations and virtual failure events as inputs and, in response, outputs virtual post-failure configurations. The failure-response simulator can be contrasted with a performance simulator that accepts a configuration and load parameters as inputs and, in response, outputs performance data.

Concise Explanation of Claim 1

Claim 1 relates to a computer system (AP1, Fig. 1, paragraph 26, page 5, lines 11-13) comprising:

a simulator (SIM, Fig. 1, paragraph 27, page 5, line 18 to page 6, line 2) including:

a virtual-failure event selector (V06, Fig. 5, paragraph 39, page 9, lines 26-27) providing for selecting a virtual-failure event corresponding to a real-failure event that applies to a real computer cluster (RCC, Fig. 1), and

a virtual-cluster generator (V04) for generating a first virtual cluster (VC1, Fig. 6) in a virtual pre-failure configuration corresponding to a real pre-failure configuration of said real computer cluster, and for, in response to selection of said virtual-failure event, generating a second virtual cluster (VC2, Fig. 7) in a virtual post-failure configuration corresponding to a real post-failure configuration that said real computer cluster would assume in response to said real-failure event.

Concise Explanation of Claim 10

Claim 10 relates to a computer-implemented method comprising:

a) generating (S4, Fig. 6, paragraph 45, page 12, lines 6-10) a first virtual computer cluster (VC1, Fig. 1) in a virtual pre-failure configuration that serves as a model for a real computer cluster (RCC, Fig. 1) in a pre-failure configuration that responds to predetermined types of failures by reconfiguring to a real post-failure configuration, said reconfiguring including migrating a real application on one real computer of said real computer cluster to another real computer of said real computer cluster;

b) selecting (S5, Fig. 6, paragraph 45, page 12, lines 7-12) a sequence of at least one of said predetermined types of failures; and

c) generating (S6, Fig. 6, paragraph 45, page 12, lines 8-10) a second virtual computer cluster (VC2, Fig. 7) in a virtual post-failure configuration that serves as a model for said real computer cluster in said real post-failure configuration.

GROUNDS OF REJECTION TO BE REVIEWED

All outstanding grounds of rejections are to be reviewed including:

The rejections of Claims 1-16 under 35 U.S.C. 103(a) for being unpatentable over U.S. Patent No. 7,107,191 to Stewart et al., "Stewart" herein, in view of U.S. Patent No. 6,393,485 to Chao et al., "Chao" herein, and further in view of U.S. Patent No. 6,137,775 to Bartlett et al, "Bartlett" herein.

ARGUMENTS

[01] Arguments for reversing the rejections of Claims 1-16 under 35 U.S.C. 103(a) for being unpatentable over Stewart in view of Chao, and further in view of Bartlett.

[02] For the purposes of these arguments, the claims are divided into two groups: Group 1 includes independent Claim 1 and its dependents Claims 2-9; and Group 2 includes independent Claim 10 and its dependents Claims 11-16.

[03] GROUP 1: CLAIMS 1-10

[04] Claim 1 requires a simulator including a virtual-failure event selector (as defined in Claim 1) and a virtual-cluster generator (as defined in Claim 1). There is agreement 1) that Stewart discloses simulators, 2) that Stewart does not disclose a virtual-failure event selector, and 3) that Stewart discloses optimizers that generate virtual-clusters, and thus qualify as virtual-cluster generators. However, Stewart's optimizers are decoupled from simulators. Therefore, Appellants disagree with the conclusion in the final action that Stewart discloses a simulator that includes a virtual-cluster generator as required by Claim 1. If there is a claim limitation of Claim 1 that is not taught or suggested by the prior art, the rejection for obviousness should be reversed, as indicated in the following MPEP section.

2143.03 All Claim Limitations Must Be Taught or Suggested

To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In*

re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). (MPEP, 2100-131 Rev. 5, Aug. 2006)

[05] Virtual-Cluster Generator

[06] The issue here, then, is whether the Examiner has established that Stewart discloses a simulator that includes a virtual-cluster generator. The Final Action points to Stewart's Figs. 1-4 (i.e., all the figures in Stewart) for the disclosure of the simulator including a virtual-cluster generator. Stewart, Fig. 1, depicts a performance simulator 104, but does not depict that it includes a virtual-cluster generator or any other sub-component. Stewart, Fig. 2, depicts a simulator 220, but does not depict that it includes a virtual-cluster generator or any other subcomponent. Stewart, Fig. 3, is a flow chart of a method including steps 324-332 which mention a simulator. However, these steps do not mention a virtual-cluster generator. Fig. 4 depicts a computer system, but does not depict a simulator, let alone one that includes a virtual-cluster generator. Thus, it appears that none of the figures in Stewart discloses a simulator that includes a virtual-cluster generator. Accordingly, the final action has failed to establish that Stewart discloses a simulator that includes a virtual-cluster generator.

[07] The final action asserts that Stewart teaches that the virtual-cluster generator generates a first virtual cluster, referring to the same four figures and column 3, lines 52-65 and column 5, lines 18-28. The passage from Stewart, column 3 reads as follows.

> FIG. **1** depicts a computer system configuration optimizer in an embodiment of the present invention. An optimizer **102** is communicatively coupled to a performance simulator **104** that predicts the performance of the resources within a computer system under defined conditions (e.g., topology and workload). In the illustrated embodiment, the conditions are defined by a topology specification and a workload definition (collectively shown as a skeleton configuration **110**), which are iteratively processed by the optimizer **102** for input to the performance simulator **104** for each simulation of the optimization. It should be understood that conditions may be specified by different or additional input information in an alternative embodiment of the present invention.

[08] This passage mentions performance simulator 104 (as depicted in Stewart, Fig. 1), but does not mention a virtual-cluster generator. The passage from Stewart, column 5, reads as follows.

> A simulator **220** represents a performance prediction system, such as that disclosed in U.S. patent application Ser. No. 10/053,731, entitled "EVALUATING HARDWARE MODELS HAVING RESOURCE CONTENTION", and U.S. patent application Ser. No. 10/053,733, entitled "LATE BINDING OF RESOURCE ALLOCATION IN A PERFORMANCE SIMULATION INFRASTRUCTURE". Generally, a simulator **220** receives topology data and workload data to simulate the performance of a computer system configuration. It should be understood that the modular architecture employed in the illustrated embodiment allows support for many different types of performance prediction systems.

[09] This passage refers to simulator 220 depicted in Fig. 2, but does not indicate that this simulator includes a virtual-cluster generator. Since none of the figures referred to and neither of the passages referred to above discloses a simulator including a virtual-cluster generator, **it is clear that the final action has failed to establish that**

Stewart teaches a simulator including a virtual-cluster generator as required by Claim 1.

[10] Virtual-Failure Event Selector

[11] The final action acknowledges that Stewart does not disclose a simulator that includes a virtual-failure event selector as required by Claim 1. The final action relies on Bartlett, Fig. 5B and column 12, lines 50-59) for a teaching of a virtual-failure event selector. The column 12 passage is descriptive of Bartlett, Fig. 5B and reads as follows.

> In step **516**, the spare capacity planning tool selects a physical span, other than the currently selected reduction span, to serve as the so-called "failed span." This failed span is the context for processing in steps **518** through **522**.
>
> In step **518**, the spare capacity planning tool renders unavailable all of the links on the failed span chosen in step **516**. For the example network **200**, the spare capacity planning tool **410** may select span **204**B as the failed span and temporarily disable its original five working links plus the five spare links allotted to it in step **508**.

[12] This excerpt discloses a spare capacity planning tool that selects a physical span and then disables its working links so that the span physically fails. While the failure is "simulated" in that it is intentionally induced, the failure is real, not virtual. Note that Bartlett does not refer to the simulated failure events as "virtual", and thus is consistent with Appellants' usage. **Since Bartlett does not disclose a virtual-failure event selector, Bartlett cannot teach modification of Stewarts' simulator to include a virtual-failure event selector.**

[13] Virtual Pre- and Post- Failure Configurations

[14] The final action acknowledges that Stewart does not disclose generating virtual clusters with the claimed pre- and post-failure configurations. As to a virtual-cluster having a pre-failure configuration, the final action relies on Chao, column 14, lines 60-67, and column 15, lines 32-47. The column 14 paragraph referred to is presented immediately below.

> Cluster Services also supports event simulation. When Recovery Services is invoked to simulate an event, it first clones the cluster configuration database. The event simulation will be performed on the private copy of the configuration database so that the original configuration database will not be affected. During a simulation, the EXECUTE statement which actually changes the state of physical resources.

[15] This paragraph does not mention a virtual pre-failure configuration at all. In fact, the terms “virtual”, “pre-failure”, and even “failure” do not appear in this passage. It should be noted that Chao does not refer to “simulation” other than in this excerpt from column 14. The paragraph from Chao column 15 that the final action relies on is presented immediately below.

FIG. 4*b* illustrates the general method **450** implemented by a multi-cluster when a node failure occurs. This method can also be applied to resource failure and resource group failure events. The group service event adapter collectively inserts exactly one node down event into the event queue (step **454**). Node_Down event processing is triggered (step **456**). Next, for every resource group that was running on the failed node, the following steps are applied (step **458**). First, recovery services compute the Next_Node for failover (step **460**). Then a decision is made if My_Node==Next_Node. If not, the system checks if there are more resource groups (step **462**). If so, then the system gets the sub-cluster to bring the specified resource group online (step **464**). If no more resource groups are available, then the system is done (step **464**). If more are available, then the system loops back to step **458**.

[16] This paragraph does not refer to a "virtual" or "pre-failure" configuration. "Failure" is mentioned, but nothing suggests that this failure is virtual instead of real. **Thus, the final action fails to establish that Chao discloses a virtual cluster with a pre-failure configuration.**

[17] The final action does not explain how the foregoing two excerpts from Chao are intended to relate. The latter paragraph refers to Chao, Fig. 4b, which is a flow chart of a method 450. This method begins with a node failure at step 452, which, in the absence of an indication to the contrary, is assumed to be a real node failure. In response, a group service event adapter inserts a Node_Down event into an event queue at 454. This triggers NODE_DOWN event processing at 456.

[18] Node_Down Event processing involves Recovery Services computing the Next-Node for failover at 460. This can result in bringing a resource group online at 464. These steps are involved in

an iterated loop that involves computing at 460 and implementing at 464. As Appellants understand the simulation described by Chao, Recovery Services can simulate the events used to reconfigure a cluster so that only the computations are involved in the iterated loop. Once all the reconfigurations are calculated, the final configuration is implemented. Using a single execution command to implement a cumulative configuration change instead of executing a series of incremental configuration changes could reduce the time required for recovery. In this interpretation, there would be a first virtual cluster with a pre-recovery configuration and a second virtual cluster with a post-recovery configuration. However, both of these virtual cluster configurations would be post-failure. In this view, Chao does not disclose generating a virtual cluster with a pre-failure configuration. **Whether or not this view is accepted, it is clear that the Final Action has not established that Chao discloses a virtual cluster with a pre-failure configuration.**

[19] Combining References

[20] Stewart discloses a performance simulator used to compare the performance of candidate virtual clusters so that an optimal configuration can be chosen for a cluster. Chao appears to disclose a configuration event simulator for decreasing the time required to recover from a real failure for a real cluster. Bartlett discloses a simulator that generates simulated real failures for checking how a real network would respond to actual (non-simulated) failures. Neither Chao nor Bartlett teaches anything that would advance Stewarts' purpose of determining an optimal configuration for a system. Accordingly, there does not appear to be any motivation for modifying Stewart in accordance with the teachings of Chao and Bartlett.

[21] Instead of providing a motivation for the proposed combination, the references teach against modifying Stewart to meet the claim limitations. For example, modifying Stewart so that the simulator rather than the optimizer generated virtual clusters would run counter to Stewart's teaching at column 3, lines 33-42, where Stewart sets forth the advantages of decoupling optimization and simulation.

[22] Conclusion for Group 1

[23] Claim 1 requires a simulator including a virtual-failure event selector and a virtual-cluster generator. The primary reference (Stewart) discloses a simulator and a virtual-cluster generator; Stewart does not disclose a virtual-failure event selector and the disclosed virtual-cluster generator is not included in the simulator. In fact, Stewart teaches that it is advantageous to decouple the simulator and the virtual-cluster generator. Bartlett implies a simulated event selector, but not a virtual-event selector. Chao teaches simulation involving virtual clusters having pre- and post-recovery configurations, but not involving a virtual cluster having a pre-failure configuration. **None of the references teach or suggest any advantage to modifying Stewart in accordance with the teachings of Bartlett and Chao, and there is no way to combine the three references to yield the present invention. For all these reasons, the rejections for obviousness of independent Claim 1 and its dependents, Claims 2-9, should be reversed.**

[24] GROUP 2: CLAIMS 10-16

[25] Virtual Pre-Failure Configuration

[26] Claim 10 involves generating a virtual computer cluster in a virtual pre-failure configuration. Stewart discloses generating virtual computer clusters, but does not disclose one with a pre-failure configuration. Chao discloses generating virtual computer clusters including those with pre- and post-recovery configurations; however, Chao does not disclose a virtual computer cluster in a pre-failure configuration. Bartlett does not disclose virtual computer clusters in any configuration. **Since none of the cited references discloses a virtual computer cluster in a virtual pre-failure configuration, there is no way combining the references can meet the Claim 10 limitation of generating a virtual computer cluster having a virtual pre-failure configuration.**

[27] Selecting a Failure Type

[28] Claim 10 requires selecting a failure type. Stewart does not address failure issues and thus does not disclose failure types or selecting failure types. Chao address a response to a real actual failure, but does not disclose selecting a failure type. Only Bartlett discloses selecting a failure type.

[29] The issue is then would it be obvious to modify Stewart's method of optimizing a configuration of a computer system to incorporate Bartlett's selection of a failure type. There is nothing in Stewart that that suggests a problem that can be addressed by such a modification. Bartlett does suggest an advantage of ensuring

restoration in an existing telecommunications network regardless of the spans that fail.

[30] Obviously, Stewart's computer system is not a telecommunications network with spans that fail, so the applicability of Bartlett to Stewart is tenuous at best. Also, Bartlett requires a real system on which to perform evaluations; testing on real systems is not practical in Stewart's context of evaluating multiple configurations to arrive at an optimal configuration. **Thus, while Bartlett teaches selecting a failure type in the context of an existing configuration of a telecommunications network, the applicability to Stewart's method of finding an optimal configuration for a computer system is questionable at best.**

[31] Virtual Post-Failure Configuration

[32] Stewart discloses virtual clusters, but not one in a virtual post-failure configuration. Bartlett does not disclose virtual clusters in any configuration. Chao teaches virtual clusters in a post failure configuration. However, Chao does not teach any advantage to the simulation outside the context of multi-cluster services. Appellant would agree that it would be obvious to modify Stewart to incorporate Chao's multi-cluster services in a network optimized using Stewart's method. However, such a modification would not necessarily involve modifying Stewart's simulator or optimizer. **Chao does not provide a motivation for modifying Stewart's simulator, so the proposed combination of references fails.**

[33] Conclusion for Group 2

[34] Even if Stewart were modified to incorporate 1) selecting failure events as taught by Bartlett, and 2) generating a virtual cluster in a virtual post-failure configuration as taught by Chao, the result would still lack a step of generating a virtual cluster in a virtual pre-failure configuration. Hence the proposed combination of references would not meet the limitations of Claim 10. Hence the rejections for obviousness of Claims 10-16 should be reversed.

[35] HINDSIGHT

[36] Throughout the arguments for rejection and response to Applicant's remarks, the final action picks and chooses from each of the references what elements it wants to combine and what elements it wants to ignore. For this reason, it appears that the Examiner has been impermissibly guided by hindsight gleaned from the present application. A brief mention of simulation with no explanation of its advantages is treated as the source of the advantages of Chao's multi-cluster system. The fact that Bartlett allows failure types to be chosen is somehow separated from the context of physical testing of a real system and treated as applicable to the activities of a performance simulator.

[37] It is a fallacy to assume that the advantages asserted in a reference can be achieved by any one of its elements by itself. Some features may not provide any advantage, or they may provide an advantage different from the one taught for the whole disclosure. The following MPEP guideline applies.

VI.PRIOR ART MUST BE CONSIDERED IN ITS ENTIRETY, INCLUDING DISCLOSURES THAT TEACH AWAY FROM THE CLAIMS

A prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention. *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984)

[38] CONCLUSION REGARDING OBVIOUSNESS

[39] Claim 1 requires a simulator that includes a virtual-failure event selector and a virtual-cluster generator that generates a virtual cluster in a virtual pre-failure configuration. None of the cited references discloses or suggests a virtual-failure event or a virtual-failure event selector. None of the cited references discloses a virtual cluster in a pre-failure configuration or a virtual cluster generator for generating such a virtual cluster. The primary reference teaches away from a simulator including a virtual-cluster generator. Clearly, the obviousness rejections of Claim 1 and its dependents should be reversed.

[40] Claim 10 requires generating a virtual cluster in a pre-failure configuration. None of the references discloses generating a virtual cluster in a pre-failure configuration. Accordingly, no combination of the cited references can yield a method meeting all limitations of Claim 10. Thus, the obviousness rejections of Claim 10 and its dependents should be reversed. **Accordingly, the rejections for obviousness of Claims 1-16 should be reversed.**

CLAIMS APPENDIX

1. *(previously presented)* A computer system comprising:

a simulator including:

a virtual-failure event selector providing for selecting a virtual-failure event corresponding to a real-failure event that applies to a real computer cluster, and

a virtual-cluster generator for generating a first virtual cluster in a virtual pre-failure configuration corresponding to a real pre-failure configuration of said real computer cluster, and for, in response to selection of said virtual-failure event, generating a second virtual cluster in a virtual post-failure configuration corresponding to a real post-failure configuration that said real computer cluster would assume in response to said real-failure event.

2. *(previously presented)* A system as recited in Claim 1 wherein, in said real pre-failure configuration, said real computer cluster runs a software application on a first computer of said real computer cluster and not on a second computer of said real computer cluster, and wherein, in said real post-failure configuration, said real computer cluster runs said application on said second computer but not on said first computer.

3. *(original)* A system as recited in Claim 1 further comprising said real computer cluster, said real computer cluster including profiling software for providing a descriptive profile of said real computer cluster, said virtual-cluster generator generating said virtual cluster in said pre-failure configuration using said descriptive profile.

4. *(original)* A system as recited in Claim 3 wherein said real computer cluster is connected to said simulator for providing said descriptive profile thereto.

5. *(original)* A system as recited in Claim 2 wherein said simulator further includes an evaluator for evaluating said virtual cluster in its post-failure configuration.

6. *(original)* A system as recited in Claim 5 wherein said simulator further includes a test sequencer, said test sequencer selecting different virtual-failure events to be applied to said first virtual cluster in said pre-failure configuration so as to result in different post-failure configurations of said virtual cluster.

7. *(original)* A system as recited in Claim 6 wherein said simulator further includes a statistical analyzer for statistically analyzing evaluations of said different post-failure configurations of said virtual cluster.

8. *(original)* A system as recited in Claim 7 wherein said test sequencer automatically tests different pre-failure configurations of said virtual cluster against different failure events, said statistical analyzer providing a determination of optimum pre-failure configuration by statistically analyzing evaluations of the resulting post-failure configurations.

9. *(original)* A system as recited in Claim 8 wherein said simulator is connected to said real computer cluster for providing said determination thereto, said real computer cluster automatically reconfiguring itself as a function of said determination.

10. *(previously presented)* A computer-implemented method comprising:

a) generating a first virtual computer cluster in a virtual pre-failure configuration that serves as a model for a real computer cluster in a pre-failure configuration that responds to predetermined types of failures by reconfiguring to a real post-failure configuration, said reconfiguring including migrating a real application on one real computer of said real computer cluster to another real computer of said real computer cluster;

b) selecting a sequence of at least one of said predetermined types of failures; and

c) generating a second virtual computer cluster in a virtual post-failure configuration that serves as a model for said real computer cluster in said real post-failure configuration.

11. *(original)* A method as recited in Claim 10 wherein steps a, b, and c are iterated for different configurations of said real computer cluster and for different sets of said predetermined failure types, said method further comprising providing a recommended configuration for said real computer cluster.

12. *(original)* A method as recited in Claim 10 further comprising:

gathering profile information about said real cluster in said first configuration, wherein said first virtual computer cluster is generated using said profile information.

13. *(original)* A method as recited in Claim 12 wherein steps a, b, and c are iterated for different configurations of said real computer cluster and for different sets of said predetermined failure types, said method further comprising providing a recommended configuration for said real computer cluster.

14. *(original)* A method as recited in Claim 13 further comprising:

transmitting said recommendation to said real computer cluster; and

implementing said recommended configuration on said real computer cluster.

15. *(previously presented)* A method as recited in Claim 10 wherein said type of failure relates to a failure of a network interface or a hard disk interface.

16. *(previously presented)* A method as recited in Claim 1 wherein said real failure event involves a failure of a network interface or a hard disk interface.

EVIDENCE APPENDIX

No evidence is being submitted with this Appeal Brief.

Related Proceedings Appendix

None. There are no related proceedings.